

CP1406 Week 5 Practical – Responsive Design

Responsive design is a website development strategy that strives to provide an optimal user experience of a website regardless of the device used. By applying responsive design principles, the webpage and content respond to the screen size of the user's device to minimise unnecessary scrolling and zooming, making reading and interacting with the site as convenient and intuitive as possible. For this week's prac, you will learn to apply responsive design principles to your websites.

For today's practical, you need to submit:

- A single zip file containing
 - The **Fitness** folder. The “Fitness” folder includes:
 - html files (index.html, template.html, contact.html, and aboutus.html) you have created for the site
 - a CSS file
 - the “images” folder contains images for the “Fitness” website.
 - The **Training Zone** folder. The “Training Zone” folder includes:
 - HTML files you have created for the site
 - a CSS file for the “Training Zone” website
 - the “images” folder contains images for the “Training Zone” website

Part 1 – Adding the Meta Viewport Element for Responsive Design

Add the `meta` viewport element to the `head` section of the Home, About Us, Contact Us, and template pages of your Fitness website. The appropriate meta viewport element helps the webpage initially load in a size that matches the viewport size.

- Open `index.html`, `about.html`, `contact.html`, and `template.html` in your text editor to prepare to modify the files.
- In all html files, just above the `<link>` tag, insert a new line.
- On the new line, add the tag
`<meta name="viewport" content="width=device-width, initial-scale=1">`
 to insert a meta viewport element for responsive design (Figure 1).
- Save all files.

```

<head>
  <title>Forward Fitness Club</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Francois+One&family=Roboto+Slab&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="css/styles.css">
</head>
<body>

```

Figure 1

Part 2 – Creating a Sticky Header

You have already created a style rule for the header element but now you will modify it for a mobile viewport and to make it appear as a fixed or sticky element. This is because the previous design uses a two-column layout for the header and the nav. Use a single-column design for a mobile viewport for a simplified design.

- With **styles.css** open in your text editor, delete the float and width declarations within the header style rule.
- Add the following styles to the **header** style rule (Figure 2)

```
/* Style rules for mobile viewport */
header {
    position: -webkit-sticky;
    position: sticky;
    top: 0;
    background-color: #000;
    height: 190px;
}

header img {
    margin: 0 auto;
}
```

- Note:** In the above code, the **position** property is used two times; once with the value **-webkit-sticky** and again with the value **sticky**. Some browsers require a prefix, such as **-webkit-**, for certain CSS properties to work.

- Save your changes.
- Open the **index.html** file in Google Chrome to view the page.
- Press the F12 key to open the Chrome developer tools.

```
/* Style rules for header */
header {
    position: -webkit-sticky;
    position: sticky;
    top: 0;
    background-color: #000;
    height: 190px;
}

header img{
    margin: 0 auto;
}
```

Figure 2

Part 3 – Editing the Navigation Style Rules for Mobile Viewports

The navigation system for a website can look different in each viewport. One of mobile design options for a navigation system is to use a single-column design, you add style rules to format the navigation list items without displaying a bullet.

- In **styles.css**, within the **nav** selector, **remove the width, float, and margin declarations**.
- Add the following code (Figure 3):

```
padding: 1%;  
margin-bottom: 1%;
```

```
/* Style rules for navigation area */  
nav {  
    padding: 1%;  
    margin-bottom: 1%;  
}
```

Figure 3

- Within the **nav ul** selector, **remove the declaration for margin**.
- Replace the **right** value of **text-align** declaration with **center** to change the text-align value (Figure 4).

```
/* Style rules for navigation area */  
nav {  
    padding: 1%;  
    margin-bottom: 1%;  
}  
  
nav ul{  
    list-style-type: none;  
    text-align: center;  
}
```

Figure 4

- Add the **ul** selector to the CSS reset style rule by adding **, ul** after the **img selector** (Figure 5). *Why is the comma important?*
- Save your changes.

```
body, header, nav, main, footer, h1, div, img, ul {
    margin: 0;
    padding: 0;
    border: 0;
}
```

Figure 5

- Locate the **nav li** selector.
- Replace the existing code** in the **nav li** selector with:

```
font-size: 1.5em;
font-family: Verdana, Arial, sans-serif;
border-top: 1px solid #ffff;
```

```
39 /* Style rules for navigation area */
40 ▼ nav {
41     padding: 1%;
42     margin-bottom: 1%;
43 }
44
45 ▼ nav ul{
46     list-style-type: none;
47     text-align: center;
48 }
49
50 ▼ nav li{
51     font-size: 1.5em;
52     font-family: Verdana, Arial,sans-serif;
53     border-top: 1px solid #ffff;
54 }
```

Figure 6

- Save your changes, and open **index.html** in Google Chrome's device mode (F12) to display the webpage in a mobile viewport.

Part 4 – Integrating a Custom Google Font

You will improve the design of the Forward Fitness Club website by integrating custom Google Fonts. Using good fonts can improve the overall design of a website. For this prac, you will use the Francois One font for the navigation links and Roboto Slab for the main content.

- Open <https://fonts.google.com/> in your browser to prepare to search for custom fonts.
- In the Search fonts text box, type **Francois One** to display the Francois One font on the webpage (Figure 7).

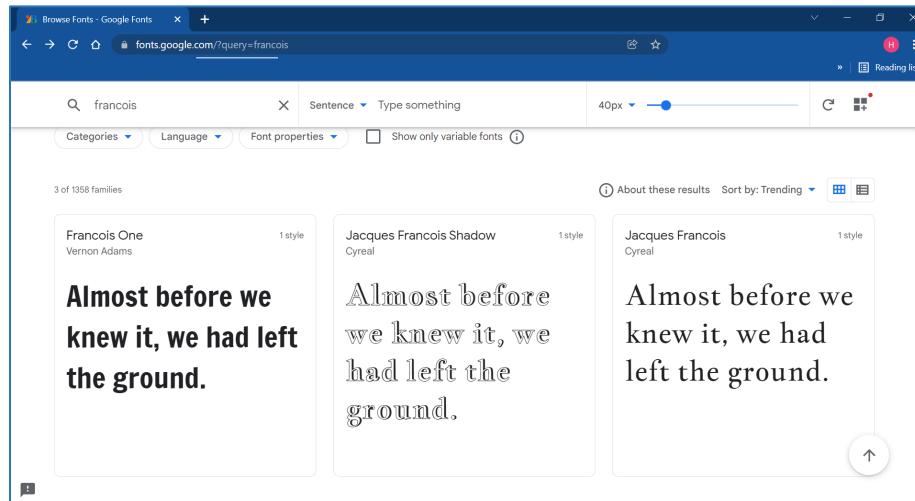


Figure 7

- Click the font, then the “Select this font” button to select the font.

We will now do the same for a second font, and then we will use both fonts at once.

- Return to the Search fonts text box, remove Francois One, and then type **Roboto Slab** to display the Roboto Slab font on the webpage.
- Tap or click the Select this font button to select the font (Figure 8).

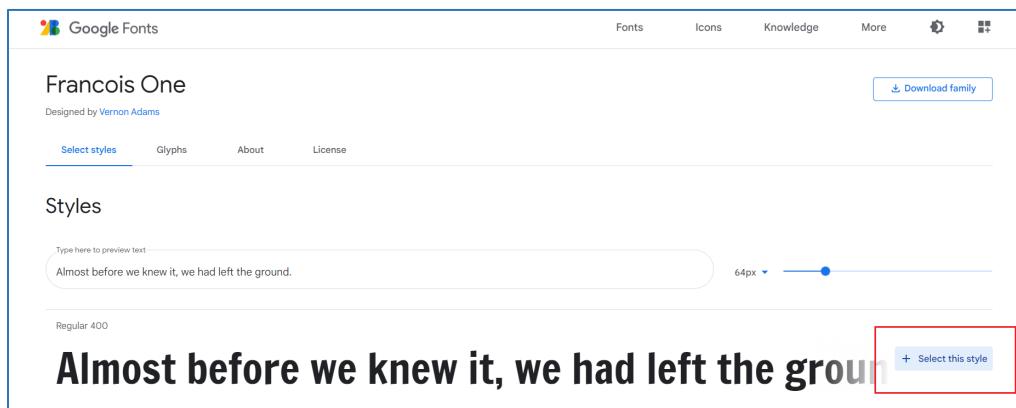


Figure 8

- Click the Families Selected collapsible window near the bottom of the webpage to expand the window (Figure 9).

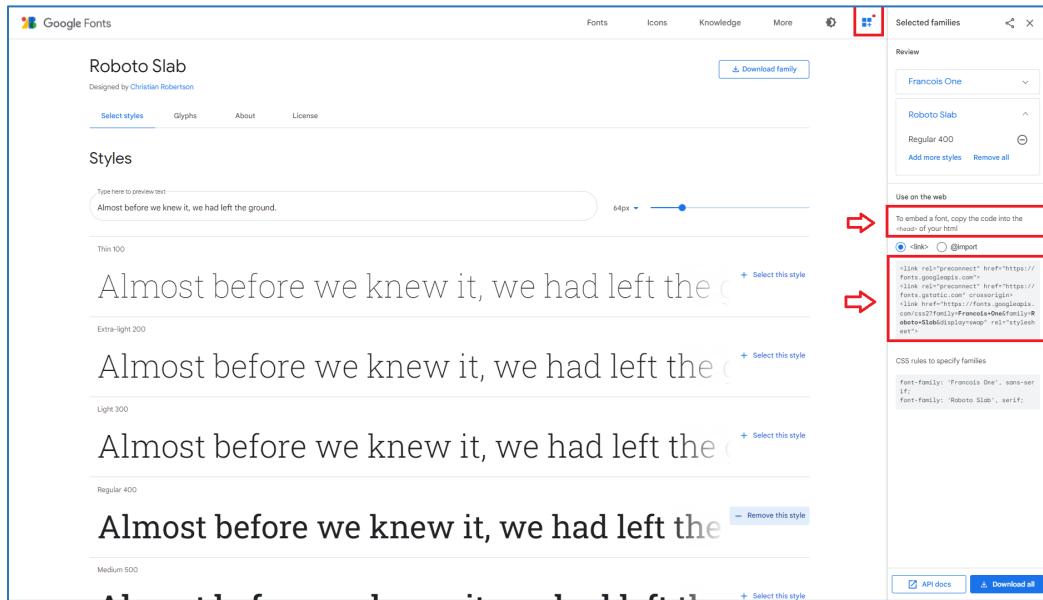


Figure 9

- Read the Embed Font instructions to understand how to embed these fonts.
- Copy the link element and keep your browser window open (Figure 9).
- Open **index.html** in your text editor, above the **<link>** tag to your style sheet, paste the link element (Figure 10).
- Save your changes.
- Repeat these steps to **about.html**, **contact.html** and **template.html** and save your changes.

```
<!doctype html>
<!-- This website template was created by: Student's First Name Student's Last Name -->
<html lang="en">
<head>
  <title>Forward Fitness Club</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://fonts.googleapis.com/css2?family=Francois+One&family=Roboto+Slab&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="css/styles.css">
</head>
```

Figure 10

- Return to your browser window to display the Google fonts webpage.
- Read the Specify in CSS instructions to understand how to apply these custom fonts in your style sheet.
- In the Families Selected window, select and copy the font-family value:
'Francois One', sans-serif;
Keep your browser window open.

- Return to your **styles.css** file in your text editor, find the font-family declaration for the **nav li** on Line 52, **remove the current value (Verdana, Arial, sans-serif;)**, and then paste the new value (Figure 11).

```
/* Style rules for navigation area */
nav {
    padding: 1%;
    margin-bottom: 1%;
}

nav ul{
    list-style-type: none;
    text-align: center;
}

nav li{
    font-size: 1.5em;
    font-family: 'Francois One', sans-serif;
    border-top: 1px solid #fff;
}
```

Figure 11

- Return to your browser window to display the Google fonts webpage.
- In the Families Selected window, select and copy the declaration:
font-family: 'Roboto Slab', serif;
- Return to your **styles.css** file in your text editor, paste the copied declaration to the **main** selector (Figure 12).
- Save your changes and open **index.html** in Google Chrome device mode to view your changes.

```
/* Style rules for main content */
main {
    clear: left;
    background-color: #fff;
    padding: 2%;
    font-size: 1.25em;
    font-family: 'Roboto Slab', serif;
}
```

Figure 12

Part 4 – Modifying the Home Page and adding Style Rules

You will modify the home page to add essential content for a mobile user. When following a mobile-first strategy, provide the mobile user with the most important, essential page content and remove or hide the desktop content. Then you will create a style rule to display the mobile class as a block element for the mobile viewport. In the mobile viewport, show the phone number with the link to make it easy for a mobile user to call the business. You do not want to display tablet or desktop content in the mobile viewport. So, you will create a style rule to hide content in the desktop class.

- Open **index.html** in your text editor to prepare to modify the page.
- Under the **<main>** tag, add the code (Figure 13).

```
<main>
<div class="mobile">

  <p>Welcome to Forward Fitness Club. Our mission is to help our
  clients meet their fitness and nutrition goals.</p>

  <h3>FREE One-Week Trial Membership!</h3>
  <p>Call Us Today to Get Started</p>
  <p class="tel-link"><a href="tel:8145559608">(814) 555-9608</a></p>

  <h4>Fitness Club Hours:</h4>
  <ul class="hours">
    <li>Mon-Thu: 6:00am-6:00pm</li>
    <li>Friday: 6:00am-4:00pm</li>
    <li>Saturday: 8:00am-6:00pm</li>
    <li>Sunday: Closed</li>
  </ul>

</div>
```

```
<!-- Use the main area to add the main content of the webpage -->
<main>

  <div class="mobile">

    <p>Welcome to Forward Fitness Club. Our mission is to help our
    clients meet their fitness and nutrition goals.</p>

    <h3>FREE One-Week Trial Membership!</h3>
    <p>Call Us Today to Get Started</p>
    <p class="tel-link"><a href="tel:8145559608">(814) 555-9608</a></p>

    <h4>Fitness Club Hours:</h4>
    <ul class="hours">
      <li>Mon-Thu: 6:00am-6:00pm</li>
      <li>Friday: 6:00am-4:00pm</li>
      <li>Saturday: 8:00am-6:00pm</li>
      <li>Sunday: Closed</li>
    </ul>

  </div>
```

Figure 13

- Add the attribute-value `class="tablet-desktop"` to the `<div id="hero">` element (Figure 14).
- For the `<div id="intro">`, change the `id` attribute to `class` and the `intro` value to `tablet-desktop` (Figure 14).

```
<!-- Hero Image -->
<div id="hero" class ="tablet-desktop">
  
</div>
```

```

</ul>

</div>
<div class="tablet-desktop">

<p>Welcome to Forward Fitness Club. Our mission is to help our
clients meet their fitness and nutrition goals.</p>

```

Figure 14

- In **styles.css**, add the following code (Figure 15).

```

/* Show mobile class, hide tablet-desktop class */
.mobile {
    display: block;
}
.tablet-desktop {
    display: none;
}

```

- Save your changes.

```

nav li a{
    display: block;
    color: #fff;
    padding: 0.5em 1em;
    text-decoration: none;
}

/* Show mobile class, hide tablet-desktop class */
.mobile {
    display: block;
}

.tablet-desktop {
    display: none;
}

```

Figure 15

You will create a style rule for the **tel-link** class to make the phone number stand out by specifying a background colour, padding, margins, width, text alignment, and rounded corners.

- Under the main selector closing bracket, add the following code to **styles.css** file (Figure 16).

```

.tel-link {
    background-color: #404040;
    padding: 2%;
    margin: 0 auto;
    width: 80%;
    text-align: center;
}

```

```

    border-radius: 5px;
}

.tel-link a {
  color: #fff;
  text-decoration: none;
  font-size: 1.5em;
  display: block;
}

```

```

main {
  clear: left;
  background-color: #fff;
  padding: 2%;
  font-size: 1.25em;
  font-family: 'Roboto Slab', serif;
}

.tel-link {
  background-color: #404040;
  padding: 2%;
  margin: 0 auto;
  width: 80%;
  text-align: center;
  border-radius: 5px;
}

.tel-link a {
  color: #fff;
  text-decoration: none;
  font-size: 1.5em;
  display: block;
}

```

Figure 16

Now you will style the hours class to add left margin. Also, for the main element, you want to slightly decrease the font size to fit more content within the mobile viewport.

- In **styles.css**, for the **main** selector, change the value for the font-size property to **1.15em** (Figure 17).
- Add the following code to your **styles.css** (Figure 17).

```

.hours {
  margin-left: 10%;
}

```

- Save your changes and open **index.html** in Google Chrome's device mode to display it in a mobile viewport.

```

77 ▼ main {
78   clear: left;
79   background-color: #fff;
80   padding: 2%;
81   font-size: 1.15em;
82   font-family: 'Roboto Slab', serif;
83 }

101 ▼ .hours {
102   margin-left: 10%;
103 }
  
```

Figure 17

Part 5 – Analysing the About Us Page for Mobile-First Design

The content on the home page is now styled for an optimum mobile viewing experience, but have a look at the About Us page. Figure 18 shows the areas that can be adjusted to better style the content for a mobile viewport.

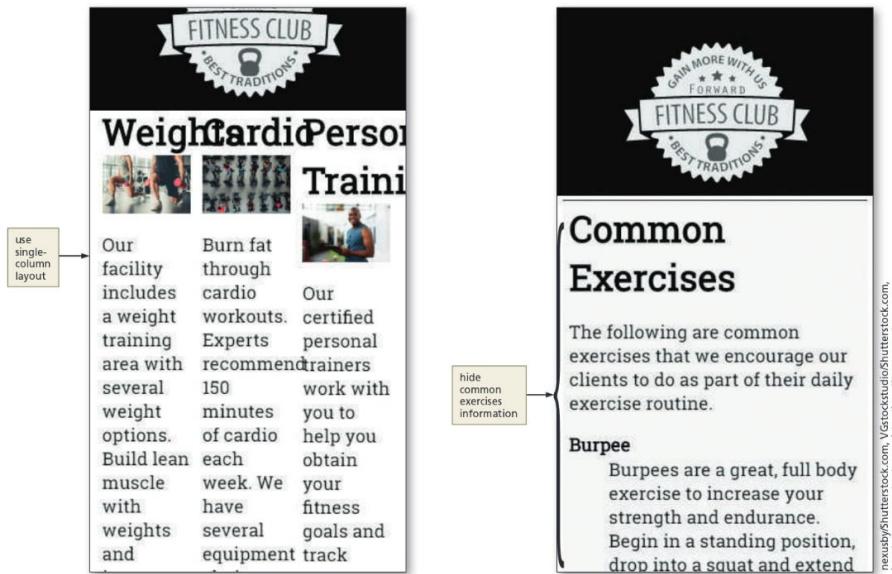


Figure 18

Part 6 – Modifying the About Us Page

We can hide nonessential content for a mobile viewport and improve the overall appearance of the page content. We will modify the HTML code on the About Us page before you modify the styles.css file so you can see how the changes you make affect the content on the webpage.

- Open **about.html** in your text editor to prepare to modify the page.

- Add


```
class="tablet-desktop"
```

 to the **unordered list element** and


```
class="round"
```

 to the **image element** as per Figure 19.
- In your **styles.css** file, add the **round** class (Figure 20):


```
.round {  
    border-radius: 8px;  
}
```

```
<div id="weights">  
  
    <h1>Weights</h1>  
     ←  
  
    <p>Our facility includes a weight training area with several weight options. Build lean muscle with weights and improve your core with weight training.</p>  
    <ul class="tablet-desktop"> ←  
        <li>Dumbbells</li>  
        <li>Kettle bells</li>  
        <li>Barbells</li>  
    </ul>  
  
</div>  
  
<div id="cardio">  
    <h1>Cardio</h1>  
     ←  
    <p> Burn fat through cardio workouts. Experts recommend 150 minutes of cardio each week. We have several equipment choices for your workout.</p>  
    <ul class="tablet-desktop"> ←  
        <li>Treadmills</li>  
        <li>Elliptical Machines</li>  
        <li>Exercise Bikes</li>  
    </ul>  
  
</div>  
  
<div id="training">  
  
    <h1>Personal Training</h1>  
     ←  
    <p> Our certified personal trainers work with you to help you obtain your fitness goals and track your progress. Personal training has many benefits.</p>  
    <ul class="tablet-desktop"> ←  
        <li>Accountability</li>  
        <li>Personalised Program</li>  
        <li>Consistent Support and Motivation</li>  
    </ul>  
</div>
```

Figure 19

Use a single column for a mobile viewport, as this will maximise the space for content. The styles for the #exercises, dt, and dd elements are not needed for the mobile viewport.

- In your styles.css file, for `#weights`, `#cardio`, `#training` selectors, remove all the declarations, except `margin: 0 2%`; (Figure 20).

```

▼ #weights, #cardio, #training {
    margin: 0 2%;
}

▼ .round {
    border-radius: 8px;
}
  
```

Figure 20

- Comment out (use the shortcut in Visual Studio Code) the style rules `#exercises`, `#excercises dt`, and `#excercises dd` (Figure 21).
- Save your changes and open `about.html` in Google Chrome's device mode to display the mobile viewport.

```

▼ /* #exercises {
    clear: left;
    border-top: 1px solid #000;
    border-bottom: 1px solid #000;
    background-color: #f2f2f2;
    padding: 1% 2%;
}

▼ #exercises dt {
    font-weight: bold;
}

▼ #exercises dd {
    padding: 0.5% 1% 2% 0;
} */
  
```

Figure 21

Your Turn:

- The **Contact Us** page includes a phone number; however, the phone number does not include a telephone link, nor is it styled for a telephone link.
- Update the page to add a telephone link and the tel-link class attribute to apply the tel-link style rule. Use the class tel-link that you created earlier to add functionality to the mobile viewport by including the telephone link. Mobile users expect this convenience on their mobile devices.
- Note:** Hyperlinks that link to a telephone number are called telephone links and work primarily on smartphones. Use an `anchor` element (`a`) to link to a telephone number by including the `href` attribute, followed by "tel:+1number" where +1 is the international dialing prefix (in this case, for the United States) and number is the phone number, including the area code. Including the international dialing prefix makes the link accurate in any location. When a user taps or clicks a telephone link from a mobile device, a dialog box is displayed, asking whether the user wants to call the phone number (see Figure 22).



Figure 22

Part 6 – Making Your Training Zone Webpages Responsive

Your Turn - Now that you have learnt and applied CSS to make your Fitness webpages responsive, you can try making your Training Zone webpages responsive. Don't forget to refer to week 1-4 pracs.

Deliverables:

- A single zip file containing
 - The **Fitness** folder. The “Fitness” folder includes:
 - html files (index.html, template.html, contact.html, and aboutus.html) you have created for the site
 - a CSS file
 - the “images” folder contains images for the “Fitness” website.
 - The **Training Zone** folder. The “Training Zone” folder includes:
 - HTML files you have created for the site
 - a CSS file for the “Training Zone” website
 - the “images” folder contains images for the “Training Zone” website